

Abstract

Instead of integrating as previously a central and global unit in one module which processes all configuration requests, now there is a plurality of hierarchically (tree structure) arranged active units which can assume this task.

A request from the lowest level (the leaves in the hierarchy) is only forwarded to the next higher level if the request could not be processed. These steps are repeated for all the levels present until the highest level is reached.

The highest level is connected to an internal or external higher level configuration memory, which contains all the configuration data required for this program run. A type of caching of the configuration data is achieved due to the tree structure of the configuration units. Access to configurations mainly takes place locally. In the most unfavorable case, a configuration must be loaded from the higher level configuration memory if the respective data is not present in any of the hierarchically arranged CTs.

Deadlocks are prevented by introducing a fixed time sequence of the configurations to be loaded and combining the configurations in a list. The status information of the CELs is saved before loading and thus remains unchanged during the processing of the entire list of configurations.